EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with Mr. Scott S. Kokka (Reg. no. 51,893) on 12/02/2009.

Discussed allowable subject matter for claims 2 and 14. Independent claim 1 needs to incorporate the limitation of claim 2, and independent claims 8 and 12 need to incorporate the limitation of claim 14. Also, "independent execution thread" in claims 1 and 8 are changed to be "virtual user thread" to be consistent with the "virtual user thread" in claim 12. Discussed 101 issues for claims 8 and 12. Applicant agreed to amend claims 8 and 12 comprising "storage" before medium to refer only CD-ROM, diskettes, and not carrier wave or other transmission medium.

3. Amend the claims according to the following paragraph:

1. (Currently amended) A method for facilitati[[o]]ng network communications from one or more ~~independent execution~~virtual user threads, comprising steps of:

storing a request object generated by one or more virtual user threads~~an independent execution thread~~ in a data structure, each of the one or more virtual user threads simulating a computer that is communicating with a server and configured to generate and then insert request objects into the data structure, the request object describing a request for an outgoing network operation and having state information;

retrieving the request object from the data structure by a dedicated worker thread, the dedicated worker thread being different from the one or more ~~independent execution~~virtual user threads and configured to process the outgoing network operation described by the request object;

modifying the request object by the dedicated worker thread to indicate a change of state;

sending a non-blocking function call from the dedicated worker thread to a networking module;

sending the request object from the dedicated worker thread to the networking module; [[and]]

executing the non-blocking function call to the networking module; and

storing a notification of completion in the data structure by the networking module, wherein the notification of completion corresponds to the request object.

2. (Canceled)

3. (Currently amended) The method of claim [[2]]1 further including the step of retrieving [[a]]the notification of completion by the dedicated worker thread from the data structure.

4. (Currently amended) The method of claim 3 further comprising the steps of:

retrieving the request object corresponding to the notification of completion by the dedicated worker thread;

determining by the dedicated worker thread whether the request described in the request object is complete; and

upon determination that the request is complete, sending a message from the dedicated worker thread to ~~the associated execution~~at least one of the one or more virtual user threads indicating that the request is complete.

5. (Previously presented) The method of claim 3 further comprising the steps of:

retrieving the request object corresponding to the notification of completion by the worker thread;

determining by the dedicated worker thread whether the request described in the request object is complete; and

upon determination that the request is not complete, sending a second non-blocking function call from the dedicated worker thread to the networking module.

6. (Currently amended) The method of claim 1 wherein the request object further includes information identifying the ~~execution~~ one or more virtual user threads.

7. (Original) The method of claim 1 wherein the data structure is a queue.

8. (Currently amended) A computer implemented system for facilitating network communications from one or more ~~independent execution~~virtual user threads, comprising:

a computer having a processor, a memory device, and a network device; and a computer-readable storage medium encoded with a computer program, the computer program comprising:

a network module configured to execute non-blocking networking function calls through the network device;

one or more request objects, each of the one or more request objects storing information describing a request for an outgoing network operation and state information;

a data structure configured to store the one or more request objects;

one or more ~~execution~~virtual user threads, each of the one or more virtual user~~execution~~ threads simulating a computer that is communicating with a server and being configured to generate and then insert the one or more request objects into the data structure; and

a worker thread, different from the one or more ~~execution~~virtual user threads, configured to retrieve a request object from the data structure, modify the state of the request object and send a non-blocking function call to the networking module, wherein the networking module is configured to store a notification of completion in a data structure configured to store the request object, the notification of completion being associated with the request object.

9. (Original) The system of claim 8, wherein the data structure is a queue.

10. (Original) The system of claim 8, wherein the worker thread is further configured to send the request object to the networking module.

11. (Currently amended) The system of claim 8, wherein the data structure is further configured to receive [[a]]the notification of completion from the networking module and store the notification of completion.

12. (Currently amended) A computer implemented system for testing a server comprising:

a computer having a processor, a memory device, and a network device; and

a computer-readable storage medium encoded with a computer program, the computer program comprising:

a networking module connected to the server over a computer network through the network device;

one or more request objects, each of the one or more request objects storing information describing a request for an outgoing network operation and state information;

one or more virtual user threads, each virtual user thread simulating a computer that is communicating with the server and configured to generate and then insert at least one of the one or more request objects into the data structure;

a worker thread, different from the one or more virtual user threads, configured to retrieve a request object from the data structure, modify the state of the request object and send a non-blocking function call to the networking module, wherein the networking module is configured to store a notification of completion in a data structure configured to store the request object, the notification of completion being associated with the request object.

13. (Original) The system of claim 12, wherein the data structure is a queue.

14. (Canceled)

15. (Currently amended) The system of claim 1[[4]]2, wherein the worker thread is further configured to retrieve the notification of completion from the data structure and to retrieve the request object associated with the notification of completion.

16. (Previously presented) The system of claim 15, wherein the worker thread is further configured to determine whether the request for the outgoing network operation is completed, based on the request object and the notification of completion.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PHUONG N. HOANG whose telephone number is (571)272-3763. The examiner can normally be reached on Monday - Friday 9:00 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyunh S. Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194

/P. N. H./
Examiner, Art Unit 2194